REMARKS

Claims 1-31 are pending in the above-referenced patent application. New claims 32 and 33 have been added by this amendment. Claims 1, 3, 4, 8, 9, 11, 12, 16, 22-25 were rejected under the judicially created doctrine of obviousness-type double patenting as being unpatenable over Claim 1 and 8 of USPN 6,198,479 and Claims 1-22 of USPN 6,288,716. Claims 1-27 were rejected under the judicially created doctrine of obviousness-type double patenting as being unpatenable over Claims 1-12 of USPN 6,466,971. Claims 28-31 were rejected as being unpatentable over USPN 5,864,669 to Osterman et al. ("Osterman") in view of USPN 6,151,624 to Teare et al. ("Teare").

Claim 25 was objected to due to certain informalities. Claims 25 has been amended to overcome the objections. No new matter has been added.

In regards to the co-pending applications referenced in the Specification, the following is an updated status:

(1) App. Ser. No. 09/104,229, has issued as Patent No. 6,288,716 B1;

(2) App. Ser. No. 09/104,298, has issued as Patent No. 6,198,479 B1;

(3) App. Ser. No. 09/104,469, has issued as Patent No. 6,243,707 B1;

(4) App. Ser. No. 09/104,606, has issued as Patent No. 6,182,091 B1; and

(5) In App. Ser. No. 09/104,297, an RCE was filed on 12/24/02, pending.

Double Patenting

Claims 1, 3, 4, 8, 9, 11, 12, 16, 22-25 were rejected under the judicially created doctrine of obviousness-type double patenting as being unpatenable over Claim 1 and 8 of USPN 6,198,479 and Claims 1-22 of USPN 6,288,716. Claims 1-27 were rejected under the judicially created doctrine of obviousness-type double patenting as being unpatenable over Claims 1-12 of USPN 6,466,971. Applicants hereby file a terminal disclaimer in compliance with 37 CFR 1.321(c) to overcome said obviousness-type double patenting rejection, as USPN 6,198,479; USPN 6,288,716 and USPN 6,466,971 are commonly owned with this patent application. Accordingly, Applicants respectfully request the withdrawal of the rejection of Claims 1-27 thereunder because the rejections are hereby rendered moot. However, if the terminal disclaimer fails to overcome the rejections, Applicants reserve the right to file a substantive response.

Claim rejections under 35 U.S.C. 103(a)

Rejection of Claims 28-31as being unpatentable over Osterman in view of Teare is respectfully traversed because the claims include limitations not taught or suggested by the references, alone or in combination.

Osterman is directed to method and system for accessing a particular instantiation of a server process, not to querying a device to obtain interface description data for command and control of that device, and storing the obtained information in a database, as claimed herein. In

col. 1, line 25 to col. 3, line 17, Osterman states that concepts defined by the RPC protocol include interfaces, endpoints, and binding handles. An interface is a description of the applications programming interfaces ("APIs") supported by a server application. A given application may have many interfaces that are accessed independently. Each instantiation, or copy, of an interface is called an endpoint. The endpoint describes a communications port that a client process may use to communicate with the server. A client process connects to a specific endpoint using a binding handle. Osterman states that in existing systems, the RPC endpoint mapper does not allow the client process to differentiate between different instantiations of a given endpoint when making a request. Instead, the endpoint mapper simply returns the first registered endpoint that corresponds to a partially-bound handle provided by the client process. In some cases, a client process may need to access a particular instantiation of an endpoint.

Osterman discloses a technique that includes annotating the endpoints and permitting a client process to select a particular process from a group of processes that are all represented by the same endpoint by selecting a particular annotation. The RPC protocol permits a server process to annotate an endpoint when the process registers the endpoint with the endpoint mapper. The annotations are used in producing human-readable displays for use, for example, in debugging or system management. However, the annotation may also be retrieved from the endpoint mapper by a client process. Thus, a server process that wishes to provide an identifiable instantiation of an endpoint does so by specifying a known annotation when registering the

endpoint. A client process then obtains a list of endpoints from the endpoint mapper and selects an instantiation having a desired annotation. The endpoint mapper acts as a registration authority for instantiations of the server process, and the annotations act as distinguishing identifiers for the instantiations.

By contrast, the present invention provides a system for performing a service on a home network by an agent querying a device to obtain application interface description data when the device is connected to the network, wherein the application interface description data includes information for commanding and controlling of the device by another device connected to the network; and storing the obtained application interface description data in a database (Claim 30).

In rejecting Claim 30, the Patent Office refers to the length passages in Osterman (col. 4, line 30 to col. 5, line 16, and col. 5, line 48 to col. 6, line 11), for the proposition that Osterman discloses the claimed limitation of an agent querying a device to obtain application interface description data when the device is connected to the network, such that the application interface description data includes information for commanding and controlling of the device by another device connected to the network. It is respectfully submitted that this interpretation of Osterman is lacking.

However, in col. 4, line 30 to col. 5, line 16, Osterman describes FIGS. 2-3 as:

> Client RPC software 135 is positioned beneath the application code 130. The client RPC software 135 receives RPCs from the application code 130 and sends them to the server process 120 over the network 125.... The server process 120 similarly includes application code 140 and server RPC software 145. The server RPC software 145 ... includes procedures that format requests received from the client RPC software 135 and pass the requests to the application code 140.... The application code 140 performs the procedure requested in the call and returns the results, if any, to the server RPC software 145. The server RPC software 145 transmits the results to the client process 115 over the network 125.
> *Referring to FIG. 3, an endpoint mapper 150 permits client application code 130 to connect to a particular interface of server application code 140, or to a particular server process (i.e., to a particular copy of an endpoint), using a partially-bound handle..... To implement the technique for identifying a particular copy of an endpoint, the server application code 140 provides annotated, partially-bound endpoints to the endpoint mapper 150 when the server application code registers with the endpointmapper (step 300).* In a simplified example, endpoints "x", "y" and "z" may be annotated, respectively, with the strings "A", "B" and "C".... Other server processes also provide endpoints (annotated or unannotated) to the endpoint mapper. The server application code 140 provides an annotated endpoint by calling the RpcEpRegister API to register the endpoint. The call includes an identifying string for the endpoint in the Annotation parameter provided to RpcEpRegister. The RPC endpoint mapper 150 stores this annotation with the endpoint and returns the annotation if the annotation is requested using RpcMgmtEpElt APIs (RpcMgmtEpEltInqBegin, RpcMgmtEpEltInqNext, and RpcMgmtEpEltInqDone). (Emphasis added).

As such, Osterman is simply describing a computer-implemented method that selects a desired copy of a particular interface in a computer system that includes a client computer and a server computer. The method includes, at the server computer, annotating the desired copy of the interface with an identifier, and, at the client computer, selecting the desired copy of the interface based on the associated identifier. The annotating and selecting steps may be implemented using the RPC protocol. There is no teaching or suggestion of the client application 130 querying the server application 140 to obtain application interface description data, such that the application interface description data includes information for commanding

and controlling of the server 140 by *another device* connected to the network, as claimed. The server application 140 of Osterman does not provide any sort of application interface description data to the other for command and control.

This is further detailed in col. 5, line 30 to col. 6, line 11, relied upon by the Patent Office, wherein Osterman states:

> To access a particular copy of a partially-bound endpoint, the client application code 130 first requests from the endpoint mapper 150 a list of fully bound endpoints corresponding to a partially-bound handle (step 305). The endpoint mapper 150 responds with a list of endpoints corresponding to the partially-bound handle, along with associated annotations (step 310)....
> The client application code 130 then selects a desired endpoint from the list of endpoints (step 315). The client application code makes this selection by scanning the annotations returned by the endpoint mapper and selecting the endpoint having a desired annotation. Finally, the client application code connects to the desired endpoint using a fully-bound handle (step 320).
> If the client wants to specify communication using a specific network transport, the client may use the RpcBindingBindingToStringBinding API and then call the RpcStringBindingParse API to determine the transport associated with a particular endpoint. If the transport of the particular endpoint does not match the desired transport, the client may move to the next endpoint on the list that is associated with the desired annotation. The client may repeat this process until an endpoint using the desired transport is found.
> The technique described above may be used, for example, to select between different copies of backup/restore processes, resource monitoring agents, or administrative support processes for common dynamic link libraries ("DLLs"). For
> example, if a software product used a single DLL called EXCHMEM.DLL to provide common heap management functionality for all services, including an Information Store, System Attendant, Message Transport Agent, and Directory Service, EXCHMEM.DLL would run in at least four different processes on every server. If a heap monitoring capability (or a heap administrative capability) is desired for this DLL, the invention could be used to allow a client application to determine which services are available, and to select a particular service to monitor.

As the above passage clearly indicates, the client application 130 receives a list of endpoints from the mapper 150 along with associated annotations. The client application 130 selects a desired endpoint from the list of endpoints by scanning the annotations returned by the endpoint mapper 150 and selecting the endpoint having a desired annotation. Then, the client application 130 connects to the desired endpoint using a fully-bound handle. Again, Osterman is simply describing selecting a desired copy of a particular interface for a client computer and a server computer. The mapper 150 annotates the desired copy of the interface with an identifier, and the client application 130 selects the desired copy of the interface based on the associated identifier. This has nothing to do with the client application 130 querying the server application 140 to obtain application interface description data, such that the application interface description data includes information for commanding and controlling of the server 140 by another device connected to the network, as claimed.

The server 140 simply provides a list of end points to the client application along with associated annotations, wherein the client application 130 selects a desired endpoint from the list of endpoints having a desired annotation. Then, the client application 130 connects to the desired endpoint using a fully-bound handle. Therefore, the server 140 is simply returning an annotated identification list of endpoints to the client 130 for the client 130 to select and connect to. As Osterman states, the endpoints are simply addresses (col. 1, lines 42-43). This is totally different than the claimed limitations of a first device providing application interface description data to a second device, whereby the first device can be commanded and controlled using the

application interface description data. Further, Osterman does not disclose the claimed step of querying another device for application interface description data. Osterman uses RPC which is totally different that querying. For at least these reasons, the Patent Office' interpretation of Osterman is respectfully traversed, and it is respectfully submitted that Osterman does not disclose the claimed limitations of querying a device to obtain application interface description data when the device is connected to the network, such that the application interface description data includes information for commanding and controlling of the device by another device connected to the network, as required by Claim 30.

Further, Osterman does not disclose the agent storing the obtained application interface description data in a database, as required by Claim 30. Not only does the mapper 150 not return application interface description data to the application 130, but also as the Patent Office states the mapper 150 does not disclose storing the annotated endpoint list in a database. Indeed, there is absolutely no reason for the mapper 150 to store the endpoint list in a data base, nor is there any motivation to do so suggested by Osterman. In addition, adding an additional storing step to the tasks performed by the mapper 150 in the middle of an RPC, would be disastrous to Osterman's system because it would substantially slow down the system's RPC performance which Osterman is seeking to improve by providing the annotated endpoint list to begin with.

Despite such factors, the Patent Office states that Teare discloses registering and storing obtained application interface data (metadata) in a database, and that incorporating Teare into the

network of Osterman would provide faster access to information. However, Teare is directed to mechanisms for associating metadata with network resources, and for locating the network resources in a language-independent manner. Owners of network resources define metadata that describes each network resource. The metadata may include a natural language name of the network resource, its location, its language, its region or intended audience, and other descriptive information. The owners register the metadata in a registry. A copy of the metadata is stored on a server associated with a group of the network resources. A copy of the metadata is stored in a registry that is indexed at a central location. A crawler service periodically updates the registry by polling the information on each server associated with registered metadata. To locate a selected network resource, a client provides the name of the network resource to a resolver process. The resolver process provides to the client the network resource location corresponding to the network resource name. Accordingly, network resources can be located merely by providing the name of the network resource in any natural language that is convenient for the client. (Abstract).

As is clear from Teare, metadata has nothing to do with interface description data that can be used by a device to command and control another device as required by Claim 30. Metadata simply describes each network resource. Teare's metadata is not stored to be used for command and control of devices. Further, the metadata is registered by an owner, not by an agent as claimed. And, there is no motivation suggested in either Osterman or Teare to combine them. Further, as detailed above, modifying Osterman as the Patent Office suggests indeed slows down

Osterman since in every RPC call the mapper 150 would have to unnecessarily store the endpoint list. Further, such a modification to Osterman would be unworkable because after the endpoint list is returned by the mapper 150 and used by the application 130, it is useless for the next RPC call and so no need to store it anywhere. If Claim 30, is once again rejected, Applicants respectfully request that the Patent Office clearly and specifically point out how such a modification to Osterman can be done to provide a working system. One of ordinary skill in the art would not look to either reference, or to combine them, achieve the features of the present invention. For the foregoing reasons, the references, alone or in combination, do not disclose or fairly suggest the limitations of Claim 30. As such, rejection of Claim 30 and all claims dependent therefrom should be withdrawn.

Claims 28 was rejection for substantially the same reasons as Claim 30. As such, for at least the reasons provided above, rejection of Claim 28 should be withdrawn.

As per Claims 29 and 31, Osterman is silent on XML. Teare mentions that the metadata can be in XML. However, this has nothing to do with the claimed limitation that an application interface description data includes XML format. As noted, the metadata and the application interface description data are totally different concepts and have different purposes and uses. For at least these reasons, rejection of Claim 29 and 31 should be withdrawn.

Further, for the above reasons, new Claims 32 and 33 should also be allowed.

Conclusion

For these, and other, reasons, Applicants believe that the claims are in condition for allowance. Reconsideration, re-examination, and allowance of all claims are respectfully requested.

Respectfully submitted,
SHERMAN & SHERMAN

9/24/03

Kenneth L. Sherman
Registration No. 33,783
2029 Century Park East
Seventeenth Floor
Los Angeles, CA 90067
Telephone: (310) 789-3200
Facsimile: (310) 789-3210

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on 9-24-03

Evelyn Menjivar
(Type or print name of person mailing paper)

(Signature of person mailing paper)

G:\KLS\SAM1\SAM10061\Amendment.9-14-03.doc